

Programmeren 2.1

Les 1.1 – Introductie

5 Sep 2022

Inhoud

- 1 Introductie
- 2 Installatie
- 3 De prompt
- 4 Interactief Python
- 5 Spyder
- 6 Jupyter Notebook
- 7 Programmerestijl
- 8 Tips voor vragen stellen

Introductie

Op BrightSpace:

- Installatie instructies
- Voorbeeld code
- Opdrachten
- Eindopdracht (t.z.t.)

Boek:

Downey, A. (2015). Think Python: How to Think Like a Computer Scientist. Version 2.4.

(<https://greenteapress.com/thinkpython2/thinkpython2.pdf>)

Voor vragen en discussie zullen we ook gebruik maken van MS Teams.

Introductie

Toetsing

Eindopdracht:

- Individueel (*)
- Beoordeeld op:
 - ▶ Correctheid
 - ▶ Programmeerstijl
 - ★ Commentaar
 - ★ Structuur
 - ★ Duidelijkheid
 - ▶ Efficiëntie

Tentamen:

- Schriftelijk
 - ▶ Geen computer
 - ▶ Geen boek
- Begrip code (wat doet dit programma)
- Kort programma op papier

(*) Elkaar om hulp vragen is uiteraard toegestaan, een tweede paar ogen kan soms helpen die rare *bug* te vinden, maar iedereen schrijft zijn eigen code. Overnemen van elkaar, of van het internet, is fraude en wordt gemeld bij de examencommissie.

Introductie

Wat is programmeren?

- Instructies geven aan een computer die stap voor stap worden uitgevoerd
- (Veel) berekeningen
- (Grote hoeveelheden) data

Een computer kan miljoenen keren simpele berekeningen uitvoeren, maar het kan niet nadenken. Dat is het werk van de programmeur.

Introductie

Waar gebruikt een toegepast wiskundige programmeren voor?

- Berekeningen
- Heuristieken in OPR
- Data visualisatie
- Data analyse
- Dashboards

Vrijwel geen enkel modern probleem in de wiskunde kan met pen en papier opgelost worden, in meeste gevallen zal er een computerprogramma geschreven moeten worden.

Introductie

Wat is programmeren?

- De menselijke taal (zelfs zoals gesproken door wiskundigen) is vaak te ambigu om door een computer begrepen te kunnen worden.
- Instructies voor een computer geven we in een **programmeertaal**, deze heeft een zeer exact beschreven structuur en grammatica die wel door een computer begrepen kan worden.
- Instructies voor een computer geschreven in een programmeertaal noemen we **code**.
- Het schrijven van code noemen we **programmeren**.

Introductie

Er zijn veel verschillende programmeertalen:

- Basic
- Pascal
- C/C++/C#
- Julia
- Go
- Fortran
- Python
- Cobol
- Java
- R
- MatLab
- IDL
- Miranda
- Haskell
- Rust

We gebruiken bij deze opleiding Python (en later R), maar de basisprincipes van programmeren zijn voor alle talen vrijwel hetzelfde. Als je eenmaal één taal hebt geleerd is het relatief makkelijk over te stappen naar een nieuwe.

Installatie

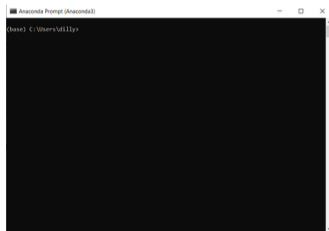
Op BrightSpace staan instructies voor het installeren van de nodige software. Dit installeert een aantal programma's:

- Anaconda Prompt (alleen windows)
- Anaconda Navigator
- Python
- Jupyter
- Spyder

De prompt

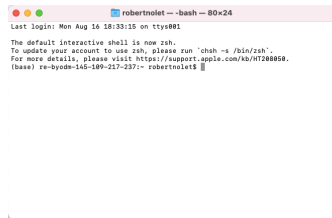
Windows

Type: "anaconda prompt" in het zoek-icoontje links onder, en voer uit.



Mac

Finder → Go → Utilities → Terminal



Het is aan te raden hier een snelkoppeling voor aan te maken.

De prompt

Wat is de prompt?

- Het maken van een grafische omgeving is moeizaam en tijdrovend.
- Veel “diepere” instellingen van je computer hebben geen grafisch menukje.
- Deze kunnen ingesteld en/of aangeroepen worden vanuit de prompt.
- Ook de programma's die we met dit vak gaan schrijven kunnen met de prompt uitgevoerd worden.

De prompt

Nuttige commando's op de prompt:

<code>dir</code>	<i>Windows</i> – Geef alle namen van bestanden in de huidige werkmap.
<code>ls</code>	<i>Mac</i> – Geef alle namen van bestanden in de huidige werkmap.
<code>cd <map-naam></code>	<i>Change directory</i> – Verander de huidige werkmap naar <map-naam>.
<code>cd ..</code>	Verander de huidige werkmap naar de omliggende map.
<code>python --version</code>	Toon het huidige versienummer van python (3.8).
<code>python</code>	Start Python op in interactieve modus.
<code>python <bestandsnaam></code>	Voer het programma in <bestandsnaam> uit.

De prompt

Nuttige commando's op de prompt:

<code>spyder</code>	Start Spyder op.
<code>jupyter notebook</code>	Start een Jupyter Notebook server op.
<code>conda list</code>	Geef een lijst van alle geïnstalleerde packages.
<code>conda install <package></code>	Installeer een package met de naam <package>.
<code>conda update <package></code>	Update een package met de naam <package>.

Interactief Python

Open de Anaconda prompt/Terminal en voer de **blauw** gekleurde commando's uit. Druk na iedere regel op enter.

```
robertnolet$ python --version
Python 3.8.8
robertnolet$ python
Python 3.8.8 (default, Apr 13 2021, 12:59:45)
[Clang 10.0.0 ] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> x = 17
>>> y = 137
>>> print(x + y)
154
>>> print(x * y)
2329
>>> z = x*x + y*y
>>> print(3*z)
57174
>>> print(y/x)
8.058823529411764
>>> quit()
```

Interactief Python

Stap voor stap:

`python --version` Geef het huidige versienummer. Dit zou 3.8.* moeten zijn.

`python` Start Python op in interactieve modus. Python commando's type je nu na de `>>>` prompt.

`x = 17` Geef `x` de waarde 17.

`y = 137` Geef `y` de waarde 137.

`print(x + y)` Bereken de waarde van $x + y$ en geef (print) het resultaat.

`print(x * y)` Bereken de waarde van $x * y$ en geef (print) het resultaat.

`z = x*x + y*y` Bereken de waarde van $x^2 + y^2$ en noem dit `z`, merk op dat deze waarde *niet* geprint wordt.

`print(3*z)` Bereken de waarde van $3z$ en print dit.

`print(y/x)` Bereken de waarde van $\frac{y}{x}$ en print dit.

`quit()` Sluit interactief python af en ga terug naar de Anaconda prompt/Terminal.

Interactief Python

Je zult merken dat je je code *exact* moet intypen, kleine afwijkingen kunnen al foutmeldingen leveren.

```
robertnolet$ python
Python 3.8.8 (default, Apr 13 2021, 12:59:45)
[Clang 10.0.0 ] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> x = 17
>>> Print(x - 49)
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
NameError: name 'Print' is not defined
>>> print(X * 78)
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
NameError: name 'X' is not defined
>>> print(3 x 8)
File "<stdin>", line 1
    print(3 x 8)
           ^
SyntaxError: invalid syntax
```


Interactief Python

Stap voor stap:

- `Print(x - 49)` Python is hoofdletter gevoelig. Het Print commando bestaat niet, het print commando wel.
- `print(X * 78)` We hebben alleen de variabele `x` gedefinieerd, niet de variabele `X`.
- `print(3 x 8)` Een vermenigvuldiging moeten we schrijven als `"*"`.

- Een `NameError` geeft aan dat er een onbekende naam (commando of variabele) staat.
- Een `SyntaxError` geeft aan dat er een onbekend symbool of structuur staat.

Interactief Python

Nog meer foutmeldingen:

```
>>> print(x * 23))
File "<stdin>", line 1
    print(x * 23))
            ^
SyntaxError: unmatched ')'
```

```
>>> print((x * 23)
...
... )
391
>>> print(x / 0)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ZeroDivisionError: division by zero
```

Interactief Python

Stap voor stap:

```
print(x * 23))
```

Een '(' gevolgd door twee ')' is sowieso fout.

```
print((x * 23)
```

Bij twee '(' gevolgd door één ')' verwacht Python dat de ontbrekende ')' mogelijk op een volgende regel komt. Het commando is nog niet “af” en moet na de “. . .” verder afgemaakt worden.

```
print(x / 0)
```

Ook voor computers is delen door nul flauwekul!

Interactief Python

```
>>> x = 17
>>> y = 23
>>> z = 30
>>> gemiddelde_xyz = (x + y + z)/4
>>> print("Het gemiddelde van 17, 23, en 30 is", gemiddelde_xyz)
Het gemiddelde van 17, 23, en 30 is 17.5
```

Ook dit is fout, maar het geeft geen foutmelding. We noemen dit een *bug*, het programma draait maar geeft de verkeerde uitkomst. We hadden door 3 i.p.v. 4 moeten delen.

We onderscheiden drie soorten fouten:

- Naam en syntax fouten – De code is niet begrijpbaar voor je computer.
 - ▶ Niet gedefinieerde (verkeerd gespelde) commandos/variabelen
 - ▶ Onbekende symbolen
 - ▶ Haakjes fouten
- Run-time fouten – De code is begrijpbaar, maar bij uitvoering gaat er iets mis.
 - ▶ Delen door nul
 - ▶ Proberen uit een niet bestaand bestand te lezen
- Bugs – De code word uitgevoerd zonder foutmeldingen, maar de code doet niet wat je wilt dat het doet

Interactief Python

Met python kan je iedere berekening uit voeren die je normaal met een rekenmachine zou doen, en veel meer!

Opdracht:

Gooi je rekenmachine in de prullenbak, en gebruik vanaf nu Python voor alle vakken bij deze opleiding.

Interactief Python

Complexere berekeningen wil je uiteraard niet één voor één aldoor intypen, je wilt ze in een bestand opslaan. Maak met kladblok (of met de Spyder editor) het bestand `abcformule.py` aan:

`abcformule.py`

```
a = 3
b = 4
c = -5
disc = b*b - 4*a*c
x1 = (-b + disc**0.5)/(2*a)
x2 = (-b - disc**0.5)/(2*a)
print("De vergelijking 3 x^2 + 4 x - 5 = 0 heeft twee oplossingen:")
print("x1 =", x1)
print("en")
print("x2 =", x2)
```

Onthoud in welke map je dit programma opslaat.

Interactief Python

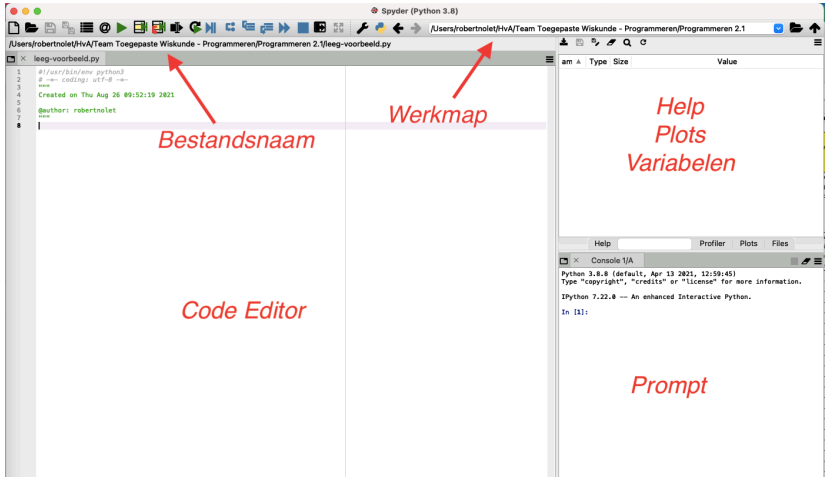
In de Anaconda prompt/Terminal typen we nu het volgende om dit programma uit te voeren

```
robertnolet$ cd /Users/robertnolet/HvA/Team Toegepaste Wiskunde -  
Programmeren/Programmeren 2.1  
robertnolet$ python abcformule.py  
De vergelijking  $3x^2 + 4x - 5 = 0$  heeft twee oplossingen:  
x1 = 0.7862996478468913  
en  
x2 = -2.119632981180225
```

- Met `cd` ga je naar de map toe waar `abcformule.py` is opgeslagen.
- Je kan dit pad “typen” door de map van je verkenner/finder naar de anaconda prompt te slepen.

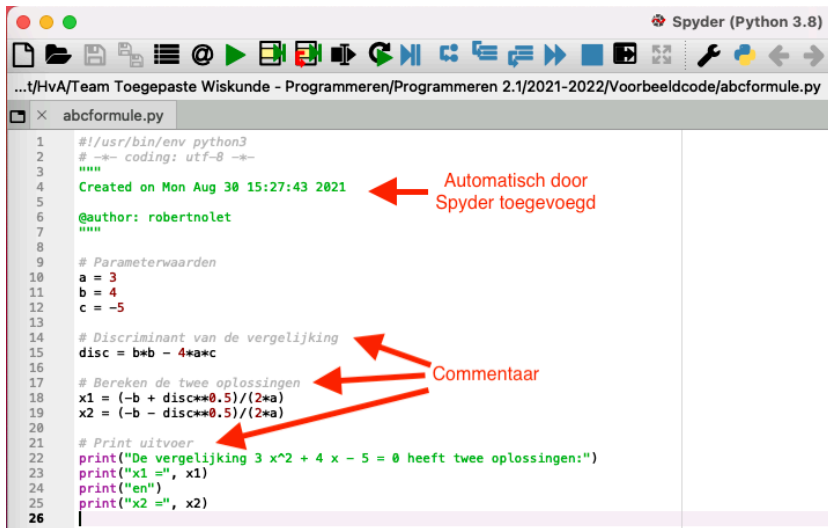
Spyder

Continu met kladblok en de prompt/terminal werken is uiteraard onhandig. Één oplossing die de functionaliteiten van dit combineert is het programma Spyder. Start dit op door in de anaconda prompt/Terminal spyder in te tikken en op enter te drukken.



Spyder

In Spyder ziet ons programma er nu zo uit:



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  """
4  Created on Mon Aug 30 15:27:43 2021
5
6  @author: robertnolet
7  """
8
9  # Parameterwaarden
10 a = 3
11 b = 4
12 c = -5
13
14 # Discriminant van de vergelijking
15 disc = b*b - 4*a*c
16
17 # Bereken de twee oplossingen
18 x1 = (-b + disc**0.5)/(2*a)
19 x2 = (-b - disc**0.5)/(2*a)
20
21 # Print uitvoer
22 print("De vergelijking 3 x^2 + 4 x - 5 = 0 heeft twee oplossingen:")
23 print("x1 =", x1)
24 print("en")
25 print("x2 =", x2)
26 |
```

Automatisch door Spyder toegevoegd

Commentaar

- Kleurcodering:
 - ▶ Getallen
 - ▶ Standaardcommando's (print)
 - ▶ Tekst (in print)
 - ▶ Commentaar
- Je kan je programma uitvoeren met de **Run** optie in het **Run** menu (snelkoppeling: F5)
- Je kan delen van je programma uitvoeren door deze code te selecteren **Run** menu → **Run selection or current line** (snelkoppeling: F9)
- De uitvoer van een *run* komt in de prompt, rechts onder.
- Variabelen en hun waardes kunnen rechtsboven ingezien worden in de **variables** tab.
- Hulp voor verschillende commando's kan rechtsboven ingezien worden in de **help** tab.

Jupyter Notebook

Een andere omgeving om Python (maar ook andere talen, zoals R) code in te schrijven is een *Jupyter Notebook*.

- Start een notebook server door `jupyter notebook` op de anaconda prompt/Terminal te typen.
- Je browser opent, met daar een overzicht van je mappen/bestanden.
- Je kan er dan voor kiezen een bestaand notebook (**.ipynb** bestanden) te openen...
- ... of een nieuw notebook aan te maken.

Jupyter Notebook

The screenshot shows a web browser window with the address bar displaying `localhost:8888/tree/HvA/Team%20Toegepaste%20Wiskunde%20-%...`. The browser tab is titled "HvA/Team Toegepaste Wiskunde". The Jupyter interface includes a top bar with "Quit" and "Logout" buttons, and a navigation bar with tabs for "Files", "Running", "Clusters", "Formgrader", "Assignments", "Courses", and "Nbextensions". Below the navigation bar, there is a text prompt "Select items to perform actions on them." and buttons for "Upload", "New", and a refresh icon. The main content area displays a file tree for the path `/ HvA / Team Toegepaste Wiskunde - Programmeren / Programmeren 2.1`. The file tree lists the following items:

	Name	Last Modified	File size
<input type="checkbox"/>	..	seconds ago	
<input type="checkbox"/>	2020-2021	8 months ago	
<input type="checkbox"/>	2021-2022	a day ago	
<input type="checkbox"/>	leeg-voorbeeld.py	19 minutes ago	113 B

Red arrows and text annotations are present on the image:

- A red arrow points to the browser tab with the label *Browser*.
- A red arrow points to the "Huidige map" (Current directory) breadcrumb path with the label *Huidige map*.
- A red arrow points to the "New" button with the label *Nieuw notebook aanmaken* (Create new notebook).
- A red arrow points to the file list with the label *Inhoud map* (Map content).

Jupyter Notebook

Dit is een markdown cel

Markdown voor shift-enter

- Markdown cellen worden gebruikt voor tekst en uitleg.
- Je typt tekst (in markdown formaat) in de cellen.
- Druk op **shift-enter** om de cel "uit te voeren" en de formatting zichtbaar te maken.

De cel hieronder is een code cel, druk up **shift-enter** om de code uit te voeren en de resultaten te laten zien.

In [5]:

```
print("Resultaten:")
print("17*37 =", 17*37)
print(123*456*789)
```

Resultaten:
17*37 = 629
44253432

Code cel

Uitvoer

Jupyter Notebook

localhost:8888/notebooks/HvA/Team%20Toegepaste%20Wiskunde... Update

Jupyter Untitled (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code Validate

Dit is een markdown cel

Markdown na shift-enter

- Markdown cellen worden gebruikt voor tekst en uitleg.
- Je typt tekst (in markdown formaat) in de cellen.
- Druk op *shift-enter* om de cel "uit te voeren" en de formatting zichtbaar te maken.

De cel hieronder is een code cel, druk up *shift-enter* om de code uit te voeren en de resultaten te laten zien.

```
In [5]: print("Resultaten:")
print("17*37 =", 17*37)
print(123*456*789)
```

Code cel

Resultaten:
17*37 = 629
44253432

Uitvoer

Jupyter Notebook

The screenshot shows a web browser window with the Jupyter Notebook interface. The browser tabs include 'HvA/Team Toegepaste Wiskunde' and 'Untitled - Jupyter Notebook'. The address bar shows 'localhost:8888/notebooks/HvA/Team%20Toegepaste%20Wiskunde...'. The Jupyter interface has a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for saving, adding cells, and running. A dropdown menu is open over the 'Add Cell' icon, showing options: 'Code' (selected), 'Markdown', 'Raw NBConvert', and 'Heading'. Two red arrows point from Dutch text to the interface: one to the 'Add Cell' icon and another to the 'Code' option in the dropdown.

Voeg nieuwe cel toe

Verander een cel van markdown/code

Jupyter Notebook

Een notebook bestaat uit cellen, er zijn twee typen cellen:

- Code cellen – Bevatten één of meer regels Python code, de uitvoer van deze code komt onder de code cel te staan.
- Markdown cellen – Bevat wiskunde, tekst, en uitleg van code en/of resultaten.


Beide typen cellen worden “uitgevoerd” door op *shift-enter* te drukken.

- Code word uitgevoerd
- Markdown krijgt formatting (*)
Bijvoorbeeld: α word omgezet naar α














(*) Jullie zullen bij dit vak niet (veel) markdown hoeven schrijven, wel zullen jullie mogelijk opdracht-notebooks krijgen waar de vraag in een Markdown cel staat en jullie de code in een code cel moeten schrijven. Wil je toch weten hoe Markdown werkt, zie dan bijvoorbeeld:

<https://www.datacamp.com/community/tutorials/markdown-in-jupyter-notebook>

Jupyter Notebook

 **jupyter** **abcformule** Last Checkpoint: a few seconds ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

           Code  Validate 

De `abc`-formule

We willen de oplossingen bepalen van de vergelijking

```
$$  
3 x^2 + 4 x - 5 = 0.  
$$
```

Dit doen we met behulp van de `abc`-formule: als

```
$$  
a x^2 + b x + c = 0,  
$$
```

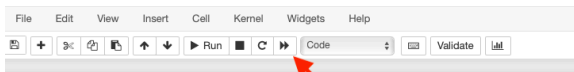
dan is

```
$$  
x = \frac{-b \pm \sqrt{b^2 - 4 a c}}{2a}.  
$$
```

```
In [ ]: # Parameterwaarden  
a = 3  
b = 4  
c = -5  
  
# Discriminant van de vergelijking  
disc = b*b - 4*a*c  
  
# Bereken de twee oplossingen  
x1 = (-b + disc**0.5)/(2*a)  
x2 = (-b - disc**0.5)/(2*a)  
  
# Print uitvoer  
print("De vergelijking 3 x^2 + 4 x - 5 = 0 heeft twee oplossingen:")  
print("x1 =", x1)  
print("en")  
print("x2 =", x2)
```

Jupyter Notebook

jupyter abcformule Last Checkpoint: a few seconds ago (autosaved)



Druk op deze knop om
alle cellen uit te voeren

De *abc*-formule

We willen de oplossingen bepalen van de vergelijking

$$3x^2 + 4x - 5 = 0.$$

Dit doen we met behulp van de *abc*-formule: als

$$ax^2 + bx + c = 0,$$

dan is

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

```
In [1]: # Parameterwaarden
a = 3
b = 4
c = -5

# Discriminant van de vergelijking
disc = b*b - 4*a*c

# Bereken de twee oplossingen
x1 = (-b + disc**0.5)/(2*a)
x2 = (-b - disc**0.5)/(2*a)

# Print uitvoer
print("De vergelijking 3 x^2 + 4 x - 5 = 0 heeft twee oplossingen:")
print("x1 =", x1)
print("en")
print("x2 =", x2)
```

```
De vergelijking 3 x^2 + 4 x - 5 = 0 heeft twee oplossingen:
x1 = 0.7862996478468913
en
x2 = -2.119632981180225
```

Jupyter Notebook

Wanneer gebruik je Jupyter vs Spyder vs prompt?

Jupyter:

- Opdrachtbladen (korte opdrachten)
- Data-analyse
- Presentaties (*)
- Online programmeren (**)
- Blogs

Spyder:

- Debuggen
- Lange / complexe programmas
- De eindopdracht (***)

Prompt:

- Het *uitvoeren* van grotere programma's met hun eigen grafische omgeving (servers, dashboards) of tools die op de achtergrond draaien.

(*) Zie: <https://medium.com/learning-machine-learning/present-your-data-science-projects-with-jupyter-slides-75f20735eb0f>

(**) <https://colab.research.google.com/> is een voorbeeld van een online Jupyter server.

(***) De eindopdracht schrijf je in Spyder maar moet vanaf de prompt uitgevoerd kunnen worden.

Programmeerstijl

Code is niet alleen een manier voor programmeurs om computers instructies te geven, het is ook hoe programmeurs onderling met elkaar praten. Programmeurs:

- Leggen begrippen aan elkaar uit met code fragmenten
- Delen code met elkaar en passen dit aan
- Werken aan grotere programma's in team verband

... en uiteraard moet je soms na een jaar weer naar een door je zelf geschreven stuk code kijken!

Doel van een goede **programmeerstijl** is om dit makkelijker te maken. Als iedereen dezelfde stijl hanteert is het makkelijker elkaars code te lezen en begrijpen, en in het algemeen zul je minder fouten maken.

Programmeerstijl

Stijl regels zijn niet bedoeld voor de computer, maar voor je medestudenten, docenten, collega's, en jezelf!

Verschillende bedrijven hebben vaak hun eigen stijl regels:

- <https://www.python.org/dev/peps/pep-0008/> (PEP 8) – Algemene stijl regels voor Python.
- <https://google.github.io/styleguide/pyguide.html> (Google) – Stijl gids voor Google programmeurs.
- Maar ook een bedrijf als Zwitserleven heeft zijn eigen stijl regels, en verwacht dat studenten die aan hun projecten werken zich hier aan houden.

Vaak heeft een bedrijf eigen software die code controleert op stijl afwijkingen.

Programmeerstijl

Stijl regels gaan o.a. over:

- Commentaar/documentatie
- Witregels en spatiegebruik
- Naamgeving variabelen en functies
- Naamgeving bestanden en mappenstructuur
- Structuur van programma

Een voorbeeld van een stijl regel uit PEP 8 is:

Limit all lines to a maximum of 79 characters.

In dit vak zullen we een aantal stijlregels hanteren, deze zullen losjes afgeleid zijn van PEP 8. Je programmeerstijl is onderdeel van de beoordeling van dit vak. Meeste lessen zullen afsluiten met een korte samenvatting van stijl regels voor de net behandelde stof.

Programmeerstijl

Alle tekst na een '#' symbool wordt door Python genegeerd. Dit is commentaar en is bedoeld om je code toe te lichten.

Stijl regel: Je code moet voorzien zijn van commentaar.

- Kies een taal, Nederlands of Engels, om je commentaar in te schrijven. Houd je hier consequent aan.
- Kort commentaar hoeft niet een volledige zin te zijn.
- Langer commentaar kan uit meerdere zinnen bestaan, met punctuatie.
- Let op duidelijkheid, correctheid, spelling en grammatica.
- Commentaar beschrijft de code niet letterlijk maar op een globaal niveau en heeft een toegevoegde waarde boven het simpelweg lezen van de code.

Programmeerstijl

Goed commentaar

```
# Deze code geeft een aantal  
# voorbeelden van een  
# goede commentaar stijl.  
  
# Grootte van dataset  
N = 500  
  
# Bereken gemiddelde  
# van x1 t/m x4.  
som_x = x1 + x2 + x3 + x4  
avg_x = som_x / 4
```

Slecht commentaar

```
# slegt comentaar  
  
# Zet N gelijk aan 500!  
N = 500  
  
# gemiddelde en som  
# van variabele  
som_x = x1 + x2 + x3 + x4  
avg_x = som_x / 4
```

Programmeerstijl

Soms voeg je ook tijdelijk commentaar toe, met dingen die je later nog aan je programma toe moet voegen.

Tijdelijk commentaar

```
# TO DO: Controleren of we niet door nul gaan delen.  
z = x / y
```

Vergeet niet je commentaar aan te passen zodra je deze “TO DO” verwerkt hebt.

Tip: Schrijf je commentaar altijd terwijl je je code aan het schrijven bent, in plaats van het achteraf toe te voegen.

Tips voor vragen stellen

Programmeren kan, voor beide beginners en experts, soms een frustrerende bezigheid zijn. Je code doet het maar niet en je komt er maar niet achter waarom! Een verkeerd geplaatst haakje kan soms ontzettend veel tijd kosten om op te sporen.

Wees niet bang om om hulp te vragen, een tweede paar ogen doet soms wonderen. Dit kan tijdens/na de les, of online op Teams.

- Probeer een zo klein mogelijk stuk code te maken dat jouw fout geeft (bijvoorbeeld door andere delen code te verwijderen door ze uit te commenten)
- Laat deze kleine code zien (je kan in een Teams chat code in een *code block* plakken.) Geef geen screenshots, iemand die jouw vraag leest moet je code kunnen copy-pasten om jouw fout te reproduceren.
- Wees specifiek:
 - ▶ Vertel welke opgave je mee bezig bent, en leg uit wat je probeert te doen
 - ▶ Geef aan welke foutmelding je krijgt, en op welke regel

En probeer ook de vragen van je medestudenten te beantwoorden!