

# Een inleiding tot Python

## Oefening 1. Inleiding tot Python

### 1.1 getallen en matrices

Start Python (Spyder). Zoek het opdrachtvenster of de console (*IPython console* in Spyder, maar het kan een andere naam hebben, afhankelijk van de omgeving) en typ de volgende commando's:

```
import scipy as sp
import numpy as np
```

*NumPy* en *SciPy* zijn twee van de vele open-source *packages* voor wetenschappelijke berekeningen die gebruik maken van de programmeertaal Python. *NumPy* en *SciPy* stellen je in staat om te spelen met arrays en matrices in Matlab stijl. Matlab is een programma met een eigen programmeertaal die speciaal is ontworpen voor wiskunde toepassingen in de academische wereld. Als je je gegevens wilt plotten moet je ook matplotlib importeren door te typen

```
import matplotlib.pyplot as plt.
```

Typ:

```
a = 4
b = 5
c = a + b
print (c)
```

De "=" wijst een waarde toe aan een variabele (verwar de toewijzingsoperator "=" niet met de gelijk-aan-operator "=="). In `derbybanan = 5`, wordt de waarde 5 toegewezen aan de variabele `derbybanan`. Je ziet dat je in Python niet het variabele type *hoeft* te benoemen bij het declareren van een variabele, wat bij C++ wel altijd moest. Naar mijn mening is het voor het begrijpen van code erg belangrijk dat je in staat bent een stuk code te vertalen in gewoon Engels, Zweeds, Nederlands of Mandarijn (of wat je moedertaal ook is).

Je kunt ervoor kiezen om code uit deze oefening te kopiëren en te plakken naar Python. Ik adviseer je dit niet te doen om twee redenen:

- 1) Het typen van commando's helpt je om de commando's te begrijpen
- 2) Microsoft Word kent verschillende aanhalingstekens, zoals "" of ". als je deze kopieert en plakt, krijg je foutmeldingen in Python die voor beginners moeilijk te begrijpen zijn.

We gaan verder met Python. Probeer `a-b`, `a/b`, `a*b`, `a**2`, `b%2` en `a+=b`. Probeer ook `np.subtract(a,b)`, `np.multiply(a,b)`, `np.divide(a,b)`, `np.power(a,b)`, `np.remainder(a,b)`, `np.add(a,b)`.

**Vraag 1.1a. Leg de volgende functies en commando's uit:**

<code>a-b</code>	.....
<code>a/b</code>	.....
<code>a*b</code>	.....
<code>a**2</code>	.....
<code>b%2</code>	.....
<code>a+=b</code>	.....
<code>np.subtract(a,b)</code>	.....
<code>np.multiply(a,b)</code>	.....
<code>np.divide(a,b)</code>	.....
<code>np.power(a,b)</code>	.....
<code>np.remainder(a,b)</code>	.....
<code>np.add(a,b)</code>	.....

Python geeft foutmeldingen als er iets fout gaat. Ze zijn niet altijd gemakkelijk te begrijpen, MAAR je moet ze altijd lezen en proberen te begrijpen. Je zult ze langzamerhand leren begrijpen en ze zijn echt nuttig. Dit is ook een goed punt om een slechte gewoonte te stoppen. Als je variabelen gebruikt, geef ze dan betekenisvolle namen. Het gebruik van betekenisvolle namen is belangrijk als je in de nabije toekomst je eigen programmering wilt begrijpen of als je wilt dat anderen het begrijpen.

Typ:

```
myfirstvector = np.array([0,1,2,3,4,5,6,7,8,9]) print(myfirstvector)
```

Typ:

```
My2ndfecktor = np.array(range(10))  
print(My2ndfecktor)
```

Typ:

```
My3dfecktor = np.arange(1,10,1)  
print(My3dfecktor)
```

Net als bij C++ maakt Python ook gebruik van arrays en lists. Als je hier meer over wilt weten, lees dan *Data Structures (pagina 76-88)* van referentie [1] (zie de lijst aan het eind van deze opdracht). Punt 6 van referentie [2] kan ook nuttig zijn. Je kunt ervoor kiezen om verder te programmeren en deze dingen later te lezen. In dit vak zullen we later lists gebruiken, maar voor nu houden we het bij arrays. Korte herhaling: een array is een variabele die meer dan één element bevat. Arrays herken je mogelijk vanuit het programmeren in C++ uit project 1, werkcollege 2. In python is het principe hetzelfde, maar de manier hoe je de instructies schrijft is anders.

Bij het maken van arrays kan het `range` commando erg handig zijn. Wanneer je commando's tegenkomt waar je nog niet eerder mee te maken hebt gehad, dan kun je in python gebruik maken van de help-functie die je meer uitleg geeft over wat functies doen. Als je in de Python console `help(range)` intypt, lees je het volgende:

```
range(...)
range([start,] stop[, step]) -> list of integers
```

```
Return a list containing an arithmetic progression of integers. range(i,
j) returns [i, i+1, i+2, ..., j-1]; start (!) defaults to 0. step is
given, it specifies the increment (or decrement). example, range(4)
returns [0, 1, 2, 3]. The end point is omitted! These are exactly the
valid indices for a list of 4 elements.
```

Als je de bovenstaande uitleg van de Python help-functie niet begrijpt, probeer dan het volgende:

```
fector = np.array(range(5,10,2))
```

In plaats van het vorige, kun je ook typen:

```
fector = np.arange(5,10,2)
```

Het effect is vergelijkbaar, maar als je grote arrays maakt, is Python 50 tot 100 sneller met het `np.arange` commando.

**Vraag 1.1b** Als je de laatste opdrachten begrijpt, maak dan een array `fector` die de eerste 10 elementen van de vermenigvuldigingstabel "tafel van drie" bevat.

```
fector = .....
```

Maak een andere array `vactur` die bestaat uit 10 elementen (naar eigen keuze). Doe dan

```
teapot = fector + vactur
print(teapot)
```

```
harddrive = fector * vactur
print(harddrive)
```

**Vraag 1.1c.** Beschrijf in woorden wat `+` en `*` in deze context doen

```
.....
.....
.....
```

Maak een nieuwe matrix door te typen

```
victor = np.array([2,3,1,5,4,3,6,8,2])
paula = victor * fector
```

**Vraag 1.1d.** Leg uit wat er gebeurde bij de uitvoering van het laatste commando.

.....  
.....  
.....

Een handige python functie is `len`. Type:

```
len(victor)
len(fector)
```

Type: `len(Victor)`

**Vraag 1.1d.** Begrijp je de foutmelding?

.....

Voordat je verder gaat, onthoud dat Python cAsE sEnSiTivE is.

## 1.2 Data plotten

Een van de voordelen van de Python console is dat je direct iets kunt berekenen. Een nadeel is dat na het stoppen van het programma of een crash, alles weg is. Als je een stuk code meerdere keren wilt gebruiken moet je het opslaan. Het opslaan van code in Python gebeurt in een .py-bestand. Op dit punt stoppen we met werken in de console (in Spyder heet dit de IPython console). In plaats daarvan zullen we beginnen met scripting. In Spyder voor de mac, ga naar de menubalk, kies File, gevolgd door New File. Dit opent een nieuw python-bestand in de editor van Spyder met de naam `untitled0.py`. Typ je code in `untitled0.py`. Wanneer je klaar bent met typen, kunt je je script opslaan en uitvoeren door op de knop F5 of op de groene pijl te drukken. Als je een script opslaat, geef het dan de naam `<goedenaam>.py` (hier kies ik `opdracht12.py`). Er zijn conventies voor namen; namen moeten beginnen met een karakter, nummer of underscore. Gebruik niet meer dan één . (punt) in een bestandsnaam.

Begin je script met:

```
import scipy as sp          # this is commenting
import numpy as np          # Python is blind for this
import matplotlib.pyplot as plt  # commenting is good
```

Maak een matrix `nicearray` beginnend met 0 en alle veelvouden van 5 tot en met 100.

**Vraag 1.2a.** Wat is het aantal elementen van deze vector?.....

Type de volgende instructie in het script:

```
carrot = np.sqrt(nicearray)
malicious = np.mean(nicearray)
```

**Vraag 1.2b.** Wat betekent `sqrt`? Wat doet `np.sqrt(nicearray)`? Wat betekent `mean`? Wat doet `np.mean(nicearray)`?

.....  
.....

`np.sqrt()` is een functie uit het `numpy` *package*. Met `import numpy as np` zie je dat `numpy` is geïmporteerd als `np`. Als we willen verwijzen naar een functie die deel uitmaakt van `numpy` moeten we `np.sqrt()` typen (je hebt eerder `np.array()` gezien, dat verwijst naar de array-functie uit het NumPy-pakket (geïmporteerd als `np`)). Als we willen verwijzen naar een functie uit het SciPy package, zoals bijvoorbeeld de functie `sin`, om sinusfuncties te maken, dan gebruiken we `sp.sin()`. Functies hebben argument(en) (hier `nicearray`) om iets in de functie te stoppen en functies kunnen iets teruggeven (hier toegewezen aan de variabele `carrot` en `malicious`). Tijdens project 4 zul je overigens je eigen functies gaan maken in Python.

Met Python en `matplotlib` kun je gemakkelijk een plot maken. Type de volgende commando's in je script:

```
plt.plot(nicearray, carrot, 'k')
plt.axis([0, 110, 0, 11])
plt.show()
```

En klik in het window boven je IPython console het tabblad "Plots" aan.

Misschien werkte de bovenstaande code niet. Als dat zo is, heeft Python u waarschijnlijk de volgende foutmelding gegeven:

**SyntaxError:** Non-ASCII character '\xe2' in file /Users/ignace/Desktop/opdracht12.py on line 14, but no encoding declared; see <http://python.org/dev/peps/pep-0263/> for details

Heb je de code uit de handleiding naar je script gekopieerd? Het stukje tekst dat je gekopieerd hebt bevat Microsoft enkele aanhalingstekens ('). De meeste softwarepakketten houden niet van Microsoft enkele aanhalingstekens. Je kunt dit probleem oplossen door de Microsoft enkele aanhalingstekens met de hand te vervangen in je script. In de editor moet het er dan uitzien als: `plt.plot(nicearray, carrot, 'k')`. De 'k' moet **groen** zijn.

Type de volgende instructie in de IPython console:

```
help(plt.plot)
```

**Vraag 1.2c.** Wat betekenen `k` en `axis([0, 110, 0, 11])`?

.....

**Vraag 1.2d.** Maak je eigen plot van de arrays `nicearray` en `carrot` met:

- 1) een gestippelde blauwe lijn met blauwe diamanten,
- 2) titel "Vierkantswortel",
- 3) x-as lopend van 0 tot 100,
- 4) y-as lopend van 0 tot 10,
- 5) en de juiste x- en y-labels.

Speel wat met de plot commando's en produceer wat kleurrijke plots. Maak de docenten trots op je! Als je meer wilt weten over plotten in python, bezoek dan

[https://matplotlib.org/users/pyplot\\_tutorial.html](https://matplotlib.org/users/pyplot_tutorial.html)

Matplotlib is een prachtig pakket. Om een idee te krijgen wat er mogelijk is met Matplotlib, bezoek <https://www.machinelearningplus.com/plots/top-50-matplotlib-visualizations-the-master-plots-python/#1.-Scatter-plot>

### 1.3 Meervoudige plots

**Vraag 1.3a.** Maak een array `xvalue` met getallen gaande van 0 tot  $2\pi$  met stappen van  $0.02\pi$ . Dit is niet eenvoudig als je niet voorbereid bent. Het probleem met de functie `range()` is dat deze geen floating-point getallen (decimalen) accepteert. De functie accepteert alleen integere/gehele getallen (getallen zoals 1, 2, 3, 112, of -1024). Als je niet bekend bent met de termen floating-point getallen en gehele getallen, voorbeelden van floating-point getallen zijn 3.12, -4.58 en 10000.1, voorbeelden van gehele getallen zijn 3, 4, -6 en 1024. Hint: gebruik `np.arange`. Hint 2:  $\pi$  wordt in Python vertegenwoordigd door `sp.pi`

```
xvalue = .....
```

Type:

```
import scipy as sp           # this is commenting
import numpy as np           # Python is blind for this
import matplotlib.pyplot as plt # commenting is good
```

```
<Here your answer to question 1.3a>
yvalue = sp.sin(xvalue)
plt.plot(xvalue,yvalue)
plt.savefig('niceplot.pdf', format='pdf')
plt.show()
```

Voorzie de vorige plot van een titel (b.v. 'heel mooi figuur') en y- en x-aslabels en maak hem mooi en kleurrijk. Als je geen hippie bent, kun je het zwart-wit laten. Je kunt gemakkelijk extra lijnen toevoegen aan je figuur. Voeg het volgende toe aan je script:

```
yvalue2 = yvalue + 0.5
plt.plot(xvalue,yvalue2)

yvalue3 = yvalue - 0.5
plt.plot(xvalue,yvalue3)
```

Je kunt ook afzonderlijke figuren plotten. Type:

```
plt.subplot(3,1,1), plt.plot(xvalue, yvalue);
plt.subplot(3,1,2), plt.plot(xvalue, yvalue2);
plt.subplot(3,1,3), plt.plot(xvalue, yvalue3);
```

Zoek uit wat de bedoeling is van de getallen tussen de haakjes van de subplot functie. Als je het niet weet, probeer dan:

```
plt.subplot(2,2,1), plt.plot(xvalue, yvalue);
plt.subplot(2,2,2), plt.plot(xvalue, yvalue2);
plt.subplot(2,2,4), plt.plot(xvalue, yvalue3);
```

Als je nog steeds geen idee hebt, probeer dan:

```
plt.subplot(3,2,1), plt.plot(xvalue,yvalue);  
plt.subplot(3,2,2), plt.plot(xvalue, yvalue2);  
plt.subplot(3,2,4), plt.plot(xvalue, yvalue2);  
plt.subplot(3,2,6), plt.plot(xvalue, yvalue3);
```

Probeer vervolgens de volgende regel code samen met de vorige 4 regels uit te voeren door te regels te selecteren in je script, en op het icoon “Run current selection or line (f9)” te drukken.:

```
plt.subplot(3,4,10), plt.plot(xvalue,yvalue);
```

**Vraag 1.3b.** Wat is de functie van de 3, de 4 en de 10 in plt.subplot(3,4,10)?

3 .....  
4 .....  
10 .....

**Vraag 1.3c.** Zal het volgende commando een foutmelding geven? Eerst denken, dan proberen

```
plt.subplot(3,2,8), plt.plot(xvalue,yvalue);
```

#### 1.4 Matrices, rijen en kolommen

Stel een script samen met (natuurlijk beginnen met `import scipy enz. )`:

```
twodimarray = np.array([[11,12,13],[21,22,23],[31,32,33]])
sumthirdrow = np.sum(twodimarray[2,])
sumfirstcolumn = np.sum(twodimarray[:,0])
```

**Vraag 1.4a.** Zoek uit wat `[2, ]` in `twodimarray[2, ]` betekent

.....

**Vraag 1.4b.** Zoek uit wat `[:, 0]` in `twodimarray[:, 0]` betekent

.....

**Vraag 1.4b.** Bereken (met een regel code) de som van de eerste kolom van `twodimarray`.

Antwoord: .....

**Vraag 1.4b.** Bereken (met een regel code) de som van de derde rij van `twodimarray`.

Antwoord: .....

**Vraag 1.4b.** Wat is `twodimarray[1,1]`?

.....

Typ (en typ echt!!! kopieer niet de enkele aanhalingstekens van Microsoft):

```
X = twodimarray[1,1] + twodimarray[2,0]
Y = np.sum(twodimarray[1,-2])
print('X = ',X,'Y = ',Y)
```

Begrijp je de eerste twee commando's? Zo niet, bestudeer dan de sectie Indexing en accessing elements (Python: slicing) van de mathesaurus.

**Ik heb de volgende literatuur en websites gebruikt om deze cursus te schrijven:**

[1] A Byte of Python, Swaroop C H.

Te downloaden op <http://www.swaroopch.com/notes/python/>

[2] <http://www.astro.ufl.edu/~warner/prog/python.html>

[3] Mathesaurus.pdf Dit is een matlab-python woordenboek.

[4] Stef Maruch Python For Dummies

[5] NumPy User Guide, Release 1.6.2

[6] SciPy Reference Guide, Release 0.11.0

[7] NumPy Reference, Release 1.6.2

Deze opdracht is een met toestemming aangepast document uit de handleiding van de eerstejaars Psychologie cursus "Programmeren" aan de Universiteit Utrecht. Met dank aan Ignace Hooge, Roy Hessels, Jeroen Benjamins, Emma Maasland en Sarah Cranko.