

Python voor Natuur- en Sterrenkundigen

Werkcollege 1, donderdag 19 november 2015

In dit werkcollege maken we kennis met Python door middel van een aantal simpele opgaven. We zullen nog niet aan de eindopdracht werken. Het is de bedoeling om de eerste opgave te maken in de interactieve interpreter en de daaropvolgende opgaven uit te werken in een editor en te draaien met behulp van de interpreter. Werk in de tweede helft van het werkcollege in ieder geval aan opgave 6 en laat je werkende code aan de assistent zien om het werkcollegepunt toegekend te krijgen.

1. *Kennismaking met de interactieve Python interpreter.* Open een terminal-venster en start de Python interpreter op door het commando `python` te geven. Doe vervolgens de volgende oefeningen.
 - (a) Maak twee variabelen `a` en `b` aan met een geheel getal of floating-point getal als initiële waarde. Voer vervolgens een aantal berekeningen uit zoals `a + b`, `a * b`, `a ** b` enzovoort. Probeer ook het resultaat op te vangen in een variabele `c`. *Tip: je kunt met "pijl omhoog" terug bij je voorgaande invoeren.*
 - (b) Wat gebeurt er als `a` een integer is en `b` een floating-point getal?
 - (c) Maak een aantal meer variabelen aan en probeer een paar langere expressies: `(a + b) * c`, `(a ** 3 * c) / d`.
 - (d) Ga na dat het optellen van een string en een getal niet werkt, maar het vermenigvuldigen van een string en een getal wel. Kun je het resultaat verklaren?
 - (e) (+) Probeer ook eens te rekenen met complexe getallen.
 - (f) Maak twee variabelen met floating-point getallen en zet deze met behulp van `print` en `.format` netjes op het scherm. Maak het eerste getal 4 vakjes breed met 1 getal achter de komma en het tweede getal 10 vakjes breed met 6 plaatsen achter de komma.
 - (g) Gegeven de string `s = "een twee drie vier vijf zes zeven acht"`. Maak met behulp van de *slice* notatie (`s[x:y]`) expressies die de woorden "twee", "vier" en "zeven" isoleren. Voorbeeld: `s[:3]` isoleert het woord "een".
 - (h) Gebruik `range` om de volgende getallenreeksen te maken: `0 t/m 9`, `1 t/m 10`, alle even getallen in `0 t/m 20` en de eerste 10 getallen van de tafel van zes.
 - (i) Om de interpreter af te sluiten gebruik je het statement `exit()`.

Zoals je hebt gezien is de interactieve interpreter een handig hulpmiddel. Bij het schrijven van een groot Python programma kun je korte statements en expressies dus in de interactieve interpreter even snel uitproberen om te kijken of deze het beoogde resultaat geven.

Voor de volgende opdrachten is het de bedoeling dat deze worden gedaan door een programma te schrijven in een editor en het daarna te draaien met de Python interpreter. Maak in je favoriete editor (bijvoorbeeld `gedit` of `kate`) een nieuw bestand: `programma.py` (of welke bestandsnaam je maar wilt). Als je wilt kun je ook kijken of de editor is ingesteld om "net" Python te schrijven: gebruik altijd spaties in plaats van tabs en spring in met 4 spaties. Schrijf vervolgens je programma. Om het programma te draaien geef je het volgende commando in de terminal: `python programma.py`.

2. Schrijf een programma dat de gebruiker om hoogte en breedte vraagt (gebruik hiervoor `int(raw_input())`). Controleer met een `if`-statement of de ingevoerde getallen groter zijn dan nul, zo niet geef een foutmelding. Reken de omtrek en oppervlakte uit en zet de resultaten op het scherm.

3. Definieer de volgende lijst bovenaan je programma:

```
l = [7, 43, 45, 32, 26, 42, 4, 33, 16, 40, 38, 39, 44, 14, 23]
```

Schrijf een loop om het gemiddelde te berekenen van deze lijst.

4. Schrijf een loop die het gemiddelde berekent van alle elementen in de lijst op een oneven positie. (Hint: gebruik `step` in `range`).
5. Vraag de gebruiker om een string in te voeren die bestaat uit een lijst gehele getallen gescheiden door spaties. Vang de invoer op in een string `s`. Je kunt de string als volgt omzetten in een lijst van gehele getallen:

```
l = map(int, s.split(" "))
```

Schrijf een loop die nu alle getallen afdrukt tussen 3 en 44 of deelbaar zijn door 6.

6. *Verplichte opdracht.* Vraag de gebruiker om een zin in te voeren. Als de zin korter is dan 10, vraag de gebruiker opnieuw. Vervang in deze zin alle klinkers met een ".". Zet op het scherm hoeveel klinkers de zin bevatte, de lengte van de string en het percentage klinkers in de string (netjes afdrukken, 1 getal achter de komma). Zet ook de nieuwe zin op het scherm. Als de string geen klinkers bevatte, geef dan de foutmelding "GEEN KLINKERS".
7. *Programmeermethoden opgavenbundel opgave 23.* Definieer bovenaan je programma een variabele `hoogte` als een positief geheel getal. Schrijf vervolgens een programma die een piramide, bestaande uit sterretjes, en ter hoogte `hoogte` afdrukt. Bijvoorbeeld voor `hoogte = 4`:

```
*
***
*****
*****
```

Tip: maak gebruik van `.format` om geen ongewenste spaties in je uitvoeren te krijgen en denk aan wat de ``-operator doet met strings!*

8. (+) Tot nu toe moesten we altijd ons programma draaien door Python expliciet aan te roepen: `python programma.py`. We kunnen er ook een echt uitvoerbaar programma van maken, net als C++ programma's. Dit gaat als volgt: in `programma.py`, voeg als de **allereerste** regel in het programma toe:

```
#!/usr/bin/env python
```

Dit heet de *she-bang regel*. Sla het programma op. Vervolgens gebruik je de terminal om de "executable-bit" te zetten op je programma: `chmod +x programma.py`. Vanaf nu kun je je programma draaien net als een C++ programma: `./programma.py`.