

Python voor Natuur- en Sterrenkundigen

Werkcollege 2, donderdag 26 november 2015

Het tweede werkcollege bestaat uit een aantal oefenopgaven en een verplichte opgave. Aansluitend kan er worden gewerkt aan de eindopdracht. Het is de bedoeling dat de opgaven worden uitgewerkt in een editor. Je kunt voor elke opgave een nieuw bestand maken. 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. Schrijf een functie `def afhalen(lijst, n)` die in een gegeven lijst de n grootste getallen opzoekt, deze in een nieuwe lijst plaatst en uit de originele lijst verwijdert. De nieuwe lijst is de returnwaarde van de functie. *Tip: Je mag gebruik maken van de functie `max`, `max(lijst)` geeft de maximumwaarde gevonden in de gegeven lijst.*

Gebruik deze functie om in de volgende lijst

```
l = [74, 48, 98, 49, 84, 58, 16, 34, 91, 3, 43, 33, 40, 30, 10]
```

de 3 grootste getallen te vinden en deze achteraan de lijst te plaatsen. Om getallen achteraan de lijst te plaatsen mag je zelf kiezen wat je wilt gebruiken: slice-assignment, de `.extend` methode of de `+`-operator toegepast op lijsten (of zelf een functie schrijven). *Optioneel: zorg ervoor dat de getallen die achteraan worden geplaatst in de volgorde van klein naar groot staan.*

2. Gegeven een lijst `l = range(20)`. Hoe kan je met 1 statement alle oneven getallen uit de lijst verwijderen? *Hint: denk aan een geavanceerde toepassing van slicing.*
3. Zie ook *Programmeermethoden opgavenbundel 2.a*. Schrijf een functie die een gegeven temperatuur in graden Fahrenheit omrekent in graden Celsius. Gebruik hiervoor

$$\text{Temp(in Celsius)} = \frac{5}{9} \times (\text{Temp(in Fahrenheit)} - 32)$$

Vervolgens gebruik je deze functie om een temperatuurschaal af te drukken (maak gebruik van een `for`-loop!):

```
Graden F:    0.0  20.0  40.0  60.0  80.0 100.0 120.0
Graden C:  -17.8  -6.7   4.4  15.6  26.7  37.8  48.9
```

4. (+) **Dictionary oefening voor de liefhebber.** Schrijf een functie die gegeven een string een histogram bepaalt met behulp van een dictionary. Zet het histogram op het scherm, bijvoorbeeld voor "helloworld":

```
e: 1
d: 1
h: 1
l: 3
o: 2
r: 1
w: 1
```

5. Schrijf een functie `streep` die een streep op het scherm zet van lengte n . Standaard wordt het `-` karakter gebruikt, maar optioneel mag ook een ander karakter worden gegeven als parameter. Test je functie met de volgende aanroepen:

```

streep(10)
streep(10, "*")
streep(20, "+")

```

6. *Verplichte opdracht.* Het doel van de opdracht is om een simpel programma te schrijven (< 40 regels) dat een lijst gehele getallen leest uit een bestand, voor elk van deze getallen faculteit berekent en de resultaten in een nette tabel met twee kolommen op het scherm zet. Zie hieronder voor een voorbeeld van het resultaat. **Structureer je programma op een nette manier, zoals besproken tijdens het hoorcollege!**

De opdracht bestaat uit een aantal onderdelen die je het beste in de volgende volgorde kunt uitvoeren:

- Schrijf een functie om $n!$ te berekenen voor een gegeven n . $n! = n \times (n-1) \times \dots \times 2 \times 1$.
- Lees het bestand `getallen.txt` in (zie website). Op elke regel van dit bestand staat een geheel getal. Zet de string die je uit het bestand leest om in een geheel getal.
- Bereken voor elk ingelezen getal de faculteit met je zelf geschreven functie. Maak voor dit getal een string " $n! = n$ ", dus bijvoorbeeld " $1! = 1$ ", en sla deze string bijvoorbeeld op in een lijst.
- We moeten nu uitzoeken hoe breed de kolommen van de tabel moeten worden. Je kunt dit doen door de grootste lengte te bepalen die voorkomt in de lijst van alle zojuist gegenereerde strings.
- Bepaal de hoogte van de tabel aan de hand van het aantal elementen in de lijst.
- Schrijf een functie om de lijst als tabel op het scherm te zetten. Je mag het volgende als "format string" gebruiken:
`"{1:{0}} | {2:{0}}".format(breedte, linker_kolom, rechter_kolom)`
 Hoe werkt dit nu precies? `{1:{0}}` drukt de eerste parameter af in een veld met de lengte opgegeven door de nulde parameter. Dus we drukken `linker_kolom` af in een veld met een lengte van `breedte`.

Voorbeeld uitvoer:

```

-----+-----
5! = 120   | 1! = 1
4! = 24    | 5! = 120
8! = 40320 | 6! = 720
6! = 720   | 4! = 24
2! = 2     | 3! = 6
-----+-----

```

- Zie ook *Programmeermethoden opgavenbundel 26*. Schrijf een functie `fibonacci(n)` die het n -de getal van Fibonacci `fibonacci(n)` berekent. Er geldt `fibonacci(1) = fibonacci(2) = 1` en `fibonacci(n) = fibonacci(n-1) + fibonacci(n-2)`, voor $n > 2$. Gebruik nog geen recursie, in plaats daarvan kun je gebruik maken van een lijst en `.append`.
- Schrijf een functie `def sorteer(lijst):` die de gegeven lijst sorteert met Bubblesort. *Tip: je kunt makkelijk omdraaien met `l[j+1], l[j] = l[j], l[j+1]`.*

Je hebt nu al genoeg vaardigheden om een goed begin te maken met de eindopdracht. Werk hier ook buiten de werkcollege-uren aan! Onderdelen waar je alvast aan kunt werken:

- Het inlezen van de bestanden `vogels.txt` en `planeten.txt`.
- Functies om een nette tabel af te drukken.
- Functies om de valversnelling voor een planeet te berekenen en om km/h naar m/s om te rekenen. (*Opmerking: Voor de andere benodigde functies heb je wat NumPy-kennis nodig.*)