

Python Programming - Labo 4

Doel van dit labo

Deze oefeningen zorgen ervoor dat je van start kan gaan met het ontwikkelen in Python en maken je vertrouwd met de basisbeginselen. In dit labo nemen we de kennis uit de voorgaande labo's mee en introduceren we lists en dictionaries

Oefening 1

Schrijf een programma dat gehele getallen van de gebruiker leest en opslaat in een list. Je programma gaat door met het lezen van waarden totdat de gebruiker 0 invoert. Dan zou het alle waarden moeten weergeven die door de gebruiker zijn ingevoerd (behalve de 0) in oplopende volgorde, met één waarde op elke regel. Gebruik ofwel de sort method of de sorted-functie om de list te ordenen.

Oefening 2

Schrijf een functie die een list en een niet-negatief geheel getal n als parameters aanneemt. De functie geeft een nieuwe kopie van de list terug met de n-aantal grootste en kleinste elementen uit de oorspronkelijke list weggefilterd. De oorspronkelijke volgorde van de elementen in de list moet niet worden behouden. Schrijf een programma dat de werking van de functie demonstreert. Je functie geeft een "foutmelding" terug als het aantal elementen in de lijst niet voldoende blijkt om het aantal grootste en kleinste elementen te doen verdwijnen. Zo bevat een lijst van 4 elementen onvoldoende elementen om de grootste en kleinste 2 elementen te doen verdwijnen. Er moet minstens 1 kunnen overblijven.

Oefening 3

Schrijf een programma dat woorden aan de gebruiker als input vraagt zolang de gebruiker geen lege lijn teruggeeft. Nadat dit laatste gebeurd is geeft het programma elk woord dat is ingevoerd exact één keer terug. De woorden moeten weergegeven worden in dezelfde volgorde als ze werden ingevoerd. Bvb als de gebruiker volgende woorden heeft ingegeven:

```
Python
Rust
Ruby
Python
JavaScript
Rust
```

Dan geeft het programma het volgende terug:

```
Python
Rust
Ruby
JavaScript
```

Oefening 4

Maak een programma dat gehele getallen inleest van de gebruiker, tot een lege lijn wordt ingegeven. Het programma toont vervolgens eerst de negatieve getallen, gevolgd door alle nullen, en tot slot gevolgd door de positieve getallen. Binnen elke groepen worden de getallen getoond in de volgorde in dewelke ze werden ingegeven door de gebruiker. Een voorbeeld: als de gebruiker de waarden 5, -2, 2 0, -1, 0, 3 en -2 ingeeft moet je programma tot volgende output komen: -2, -1, -2, 0, 0, 5, 2 en 3. Je programma print elke waarde uit op een eigen regel.

Oefening 5

Als we een lijst met elementen in een tekst weergeven scheiden we de elementen doorgaans met een komma. Voor het laatste element komt dan een “en”.

Voorbeelden:

```
rozen
rozen en kranten
rozen, kranten en ketels
rozen, kranten, ketels en wanten
```

Schrijf een functie die een list van strings als enige parameter aanneemt. Jouw functie geeft een string terug die alle elementen van de list geformatteerd teruggeeft op de manier die je hierboven ziet. Je test de functie in je programma uit met lists van verschillende lengtes.

Oefening 6

Om de hoofdprijs in een bepaalde loterij te winnen, moet men alle 6 nummers op een loterijbriefje matchen met de 6 nummers tussen 1 en 49 die zijn getrokken door de organisator van de loterij. Schrijf een programma dat een willekeurige selectie van 6 nummers genereert voor een loterijbriefje. Zorg ervoor dat de 6 geselecteerde nummers geen dubbele nummers bevatten.

Geef de nummers in oplopende volgorde weer.

Oefening 7

Een sublist is een list die deel is van een grotere list. Een sublist kan een list zijn die bestaat uit een enkel element, meerdere element of zelfs geen enkel element.

Bijvoorbeeld: [1], [2], [3] en [4] zijn allen sublists van de list [1, 2, 3, 4]. De list [2, 3] is eveneens een sublist van [1, 2, 3, 4], maar [2, 4] is geen sublist van [1, 2, 3, 4] omdat elementen 2 en 4 niet aangrenzend zijn. Een lege list is een sublist van elke list. Dus [] is een sublist van [1, 2, 3, 4]. Een list is ook een sublist van zichzelf, dus [1, 2, 3, 4] is ook een sublist van [1, 2, 3, 4].

Schrijf een functie `is_sublist` die twee lists (een kleinere en een grotere) als parameters aanneemt en bepaalt of de ene list een sublist is van de andere. De functie geeft `True` terug als de kleinere list een sublist is van de grotere. In je programma demonstreer je de werking van de functie.

Oefening 8

Gebruik makend van de definitie van een sublist uit de vorige oefening schrijf je een functie die een list teruggeeft met daarin alle mogelijke sublists van een list die aan de functie als argument wordt meegegeven. Bijvoorbeeld: de sublists van [1,2,3] zijn [], [1], [2], [3], [1, 2], [2, 3] en [1, 2, 3]. Merk op dat je functie altijd een list teruggeeft die minstens een lege list teruggeeft omdat een lege list een sublist is van elke list. Voeg een programma toe die de werking van de functie illustreert door de functie verschillende keren aan te roepen met verschillende lists.

Oefening 9

Schrijf een functie `omgekeerd_zoeken` die alle keys in een dictionary vindt die een bepaalde value hebben. Deze functie heeft twee parameters: de dictionary en de waarde die moet worden gevonden. De functie geeft een (mogelijke lege) list van

keys terug uit de dictionary die dus mappen op de gezochte waarde. Je voegt een programma toe die de functie een aantal keer demonstreert met relevante voorbeelden.

Oefening 10

In deze oefening simuleer je 1000 worpen van twee dobbelstenen. Je schrijft een functie die geen parameters aanneemt en de worp van twee dobbelstenen simuleert. Deze functie geeft het totaal van de worpen terug. Schrijf een programma dat deze functie 1000 keer uitvoert. Het programma telt het aantal keer dat een bepaalde waarde voorkomt. Je gebruikt een dictionary om de data bij te houden. Na afloop toont het programma een tabel die de data samenvat op de volgende manier:

Totaal	Gesimuleerd Percentage	Verwacht Percentage
2	2.90	2.78
3	6.90	5.56
4	9.40	8.33
5	11.90	11.11
6	14.20	13.89
7	14.20	16.67
8	15.00	13.89
9	10.50	11.11
10	7.90	8.33
11	4.50	5.56
12	2.60	2.78

Oefening 11

Maak een programma die het aantal unieke karakters bepaalt en weergeeft in een string die door de gebruiker werd ingegeven. Bijvoorbeeld: "Ik hou van Python!" heeft 13 unieke karakters. Gebruik een dictionary om dit probleem op te lossen.

Oefening 12

We noemen twee woorden anagrammen als ze dezelfde letters bevatten, maar in een verschillende volgorde. Schrijf een programma dat twee strings van een gebruiker inleest en bepaalt en teruggeeft of beide strings anagrammen zijn. Gebruik een dictionary om dit probleem op te lossen.

Oefening 13

Anagrammen kunnen ook uit meerdere woorden bestaan. Bijvoorbeeld: “Marten Asmodom Vilijn” en “Mijn naam is Voldemort” als je hoofdletters en spaties niet meerekent. Breid oefening 12 uit zodat je programma kan checken of twee zinnen anagrammen zijn. Je negeert hierbij hoofdletters, punctuatie en spaties.

Oefening 14

Een bingo-kaart bestaat uit 5 kolommen van telkens 5 getallen. De kolommen zijn gelabeld met de letters B, I, N, G en O. Er zijn 15 getallen die onder elke letter kunnen terecht komen:

- Onder de “B” zijn dat 1 tot en met 15
- Onder de “I” zijn dat 16 tot en met 30
- Onder de “N” zijn dat 31 tot en met 45
- Onder de “G” zijn dat 46 tot en met 60
- Onder de “O” zijn dat 61 tot en met 75

Schrijf een functie die een willekeurige bingo-kaart aanmaakt en deze als dictionary opslaat. De keys zijn de letters, de waarden telkens een list van 5 getallen. Schrijf een tweede functie die de bingo-kaart weergeeft met de gelabelde kolommen en een dictionary aanneemt als parameter.

Oefening 15

In het spel Scrabble heeft elke letter een bepaalde waarde. De totale score van een woord is de som van de scores van elke letter. Schrijf een programma dat de Scrabble-score voor een woord weergeeft. Maak een dictionary die de letters mapt op een letterwaarde. Gebruik vervolgens deze dictionary om de score te berekenen. Alle letterwaarden op een rij van scrabble zijn: A=1, B=3, C=5, D=2, E=1, F=4, G=3, H=4, I=1, J=4, K=3, L=3, M=3, N=1, O=1, P=3, Q=10, R=2, S=2, T=2, U=4, V=4, W=5, X=8, Y=8, Z=4.