

Start to Program: Python, Les 8

List Comprehensions

Je kan een lijst automatisch aanvullen in Python. Tot nu toe heb je met behulp van een iteratie een lijst aangevuld. Telkens als er door de loop werd gegaan voeg je dan een extra item toe.

Dit kan ook korter. Wanneer de variabele wordt aangemaakt kan er meteen een loop worden toegekend.

Hiervoor heb je een 'telbaar' item nodig. Bijvoorbeeld een andere lijst of een string. Standaard ziet de code er als volgt uit.

```
lijst_var = [i for i in telbaar_item]
```

lijst_var => variabele waarin de lijst komt

i => index van de for-loop teller

telbaar_item => variabele of datatype dat telbare delen heeft.

Praktisch:

```
naam_letters = [letter for letter in 'Kristoff']  
print(naam_letters)
```

Dit geeft hetzelfde resultaat:

```
naam_letters = []  
  
for letter in 'Kristoff':  
    naam_letters.append(letter)  
  
print(naam_letters)
```

Met Conditie

Je kan ook condities toevoegen aan de iteratie. Zo kan er een nieuwe lijst gemaakt worden zonder een bepaald item.

```
weekdagen = ['maandag', 'dinsdag', 'woensdag', 'donderdag', 'vrijdag']  
  
dagen = [dag for dag in weekdagen if dag != 'woensdag']  
print(dagen)
```

Wiskundige operatoren

Er kunnen ook rechtstreeks wiskundige berekeningen gemaakt worden. De resultaten worden dan meteen in een lijst opgeslagen.

```
getal_lijst = [x**2 for x in range(10)]  
print(getal_lijst)
```

Voeg nu een conditie toe om alleen de even getallen op te slaan.

Je kan ook meerde loops of geneste loops opslaan. Stel je wil 2 lijsten vermenigvuldigen.

```
x_getal = [20,40,60]  
y_getal = [2,4,6]  
vermenigvuldig = [x*y for x in x_getal for y in y_getal]  
Dit geeft hetzelfde resultaat:  
vermenigvuldig = []  
for x in x_getal:  
    for y in y_getal:  
        vermenigvuldig.append(x*y)
```

Opdrachten:

1. Maak een lijst met alle getallen tussen 50 en 100.
2. Maak een lijst met alle even getallen tussen 50 en 100
3. Maak een lijst met getallen tussen 1500 en 2700 die deelbaar zijn door 5 of door 7.
4. ['abc', 'xyz', 'aba', '1221', '22', 'cedc', 'aa']
Maak een nieuwe lijst met de elementen die langer zijn dan 2 tekens en het eerste en laatste teken gelijk zijn.
5. `getallen = [0,1,2,3,4,5,6,7,9]`
Vul een lijst met elementen waarvan de INDEX deelbaar is door 3.

Datatype: Tuples

Een andere soort van lijsten. Tuples en lists hebben 1 groot verschil. Tuples zijn lijsten die niet aangepast kunnen worden.

Dus eenmaal de waarden in de tuple vastliggen blijven ze zo. Je maakt hiermee duidelijk in je code dat je geen veranderingen aan deze lijst wil.

Tuples worden op een gelijkaardige manier aangemaakt. Het verschil hier zit in het soort haakje er gebruikt wordt.

```
tuple_var = ()
```

Geeft een lege tuple.

```
weekdagen = ('maandag','dinsdag','woensdag','donderdag','vrijdag')
```

Geeft een tuple met de weekdays. Deze kan verder niet meer worden veranderd.

Indexering Tuples

De indexering werkt net zoals bij strings en lijsten. Net zoals de opvraging van deze indexering.

```
weekdagen = ('maandag','dinsdag','woensdag','donderdag','vrijdag')
print(weekdagen[2])
OF
print(weekdagen[-3])
```

Slicing/opdelen

Terwijl de waarden van de tuple niet veranderlijk zijn kan je de tuple wel opdelen.

```
print(weekdagen[1:3])
print(weekdagen[:4])
print(weekdagen[1:])
```

Je kan ook een lijst omdraaien.

```
print(weekdagen[::-1])
```

Optellen en vermenigvuldigen

Je kan tuples samen voegen om meerdere malen 'printen' net zoals lijsten.

```
weekdagen = ('maandag','dinsdag','woensdag','donderdag','vrijdag')
weekenddagen = ('zaterdag','zondag')

dagen = weekdays + weekenddagen
```

tuple functies

lengte van een tuple opvragen.

```
print(len(weekdagen))
```

max en min

Als je een tuple van numbers hebt kan je het grootste of kleinste number opvragen.

```
getallen = (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)
print(max(getallen))
print(min(getallen))
```

Zoals je merkt werken de meeste methodes hetzelfde als bij gewone lijsten.

Het grote verschil is dus dat een lijst aanpasbaar is en een tuple niet.

```
weekdagen[0] = 'zondag'
```

Dit zal dus een error geven.

Om dus de lijst aan te passen moet er veranderd worden van datatype.

Tuple naar list of list naar tuple.

Hiervoor kan je comprehension gebruiken om elk item in een nieuwe list of tuple op te slaan.

Opdracht:

Maak van de tuple weekdays een list.

Maak van de tuple weekdays een list zonder woensdag