

Lijstbegrippen (Comprehensions)

Naast sequentiebewerkingen en lijstmethoden, bevat Python een meer geavanceerde bewerking die lijstbegrip wordt genoemd.

Lijstbegrippen stellen ons in staat om lijsten op te bouwen met een andere notatie. Je kunt het beschouwen als in wezen een eenregelige for-lus die tussen haakjes is gebouwd. Voor een eenvoudig voorbeeld:

Voorbeeld 1

```
# Grijp elke letter in een string
lst = [x for x in 'word']

# Controleren
lst

['w', 'o', 'r', 'd']
```

Dit is het basisidee van een lijstbegrip. Als u bekend bent met wiskundige notatie, zou dit formaat bekend moeten zijn, bijvoorbeeld: $x^2 : x \in \{0, 1, 2, \dots, 10\}$

Laten we nog een paar voorbeelden bekijken van lijstbegrippen in Python:

Voorbeeld 2

```
# Vierkante getallen binnen range (bereik) en veranderen in lijst
lst = [x**2 for x in range(0,11)]

lst

[0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

Voorbeeld 3

Laten we eens kijken hoe we if-instructies kunnen toevoegen:

```
# Check for even numbers in a range
lst = [x for x in range(11) if x % 2 == 0]

lst

[0, 2, 4, 6, 8, 10]
```

Voorbeeld 4

Kan ook ingewikkelder rekenen:

```
# Converteer Celsius naar Fahrenheit

celsius = [0,10,20.1,34.5]

fahrenheit = [((9/5)*temp + 32) for temp in celsius ]

fahrenheit
[32.0, 50.0, 68.18, 94.1]
```

Voorbeeld 5

We kunnen ook geneste lijstbegrippen uitvoeren, bijvoorbeeld:

```
lst = [ x**2 for x in [x**2 for x in range(11)]]
lst
[0, 1, 16, 81, 256, 625, 1296, 2401, 4096, 6561, 10000]
```

Later in de cursus zullen we leren over generatorbegrippen. Na deze lezing zou u zich op uw gemak moeten voelen bij het lezen en schrijven van elementaire (basis) lijstbegrippen.